Workshop report on "New Technologies for Chess"

T. Anthony Marsland
University of Alberta
Edmonton

As reported in the December 1988 issue, the Nineteenth NACCC was held at the recent Supercomputing Conference, Orlando, Florida. As part of the event, a successor to the previous years' workshop[3] was held. The theme reflected both the nature of the hosting conference and the continuing impact of technology on the improved performance of chess programs.

The moderator opened the session by briefly reviewing the highlights of the previous workshop, drawing the listener's attention particularly to the contributions of those who were not present this year. He cited the work of Ken Thompson, who had described a hardware-software system for reliably transcribing game listings in chess books into machine readable form, thus making it possible to correct and improve standard reference works. Another contributor, Danny Kopec and his group at the University of Maine, explained how they brought expert system techniques to bear on the analysis of difficult standard endgames. Finally, Burt Wendroff's interesting talk on problems with the standard implementation of transposition tables was recalled. An extended version of that talk appeared in a later issue of the journal[6].

The new contributions opened with a presentation by Feng-hsuing Hsu of Carnegie-Melon University, who assessed the needs of chess machines in the final years before they become International Grandmasters. He is convinced that it is possible to build a machine on a single chip and use it to search at a rate of 3 million nodes per second. To this he adds a class of parallel branch and bound algorithms (of which the alpha-beta algorithm is but a special case), that can achieve a 300-fold speedup with a thousand processors. Still, more chess-specific work needs to be done, especially in the area of opening book preparation, evaluation function tuning, and in transforming into won endgames. However, he noted that Belle achieved the Fredkin Master Prize with a program that searched 160,000 nodes per second; Deep Thought made the Fredkin Grandmaster norm (ELO of 2500) when searching 700,000 nodes per second. How much more can it possibly take to beat the world champion? In a brute force sense then, the predictions may be enough, but the near future also offers improved search extensions[1], and other dynamic expansions exclusive to forced situations. These selective extensions are thought to be the main arsenal of human players to keep their thoughts narrowly focussed. To support his view he quoted Deep Thought's success in finding a 19 move mate, by far the deepest non-forced mate found so far.

The next speaker, Robert Hyatt of the University of Alabama, reviewed some of the results from his newly completed Ph.D. thesis. That work was mainly concerned with high performance parallel algorithms for depth first search. He concentrated on the problem of searching variable depth trees and acknowledged that the toughest problem was the balancing of the search load across the available processors. This is a technically difficult task to program. Essentially the methods used involve dynamic tree splitting, but without a master/slave relationship in the processors. The aim is to ensure that processors are only idle when there is absolutely no work to be done. This thesis will no doubt spawn some articles for future issues of the journal.

Another speaker also dealt with parallelization for chess. Ed Felton, one of the authors of Waycool, is convinced that high parallelism is possible in practice. The key is dynamic load balancing to reassign idle processors to the search of active parts of the tree. This also requires instantaneous communication of new alpha-bounds as soon as they are found. One major feature of Waycool is its distributed hash table. Each processor in the N-cube handles one-Nth of the table (here, each processor handles 2K entries). Thus with a good hash key no bottlenecks arise. Also as processors are added, the table automatically increases in size. The search strategy is not unlike that described by Robert Hyatt, which in turn drew heavily on a long tradition of parallel methods[2,4,5]. One important controversy arose when the speaker claimed 105-fold speedup with 250 processors, and 170-fold speedup with 512. After some discussion it became clear that the traditional measure of speedup, comparison to the best available uniprocessor, was not used. His basic system consisted of one processor with 1 unit (512 kilobytes) of memory, but naturally the N-processor system had N units of memory, distributed equally across the processors. Nevertheless, speedup measures require that the effect of the processors be isolated from extra memory and other considerations, and so the comparison should have been made to one processor with N units of memory. On the other hand, the approach in Waycool is an excellent example of the advantages of distributed computation, since it might be impossible for one processor to address N units of memory directly.

As always, Tony Scherzer tried to be controversial. At the previous workshop he had described an experiment in rote learning; here he wondered aloud, why in multiprocessor systems are all the processors identical, and what it might take to compute at the rate of 3 million nodes per second? Certainly only a three or four operation ALU is needed to do the essentials of the alpha-beta bound test - although optimizations here might be important only if that test formed the dominant computation. It seems clear that special purpose hardware can speed the evaluation function, even though "lazy evaluation" and "sufficient calculation" tests reduce the computational demands in most programs. Nevertheless, the time is ripe for original work in special hardware for position evaluation.

The workshop closed with brief presentations by two speakers. First, Jonathan Schaeffer reviewed some performance results from Para-Phoenix, and then considered how Grandmasters come up with long-range plans; plans which encapsulate David Levy's approach to defending against chess programs "do nothing, but do it carefully". This was followed by some words to a newcomer to the competition, Martin Hirsch, author of AI Chess. That program held its own against the strongest competition to date, showing that original ideas and the development of error-free code remain the keys to success in all programming endeavours.

To wrap-up the workshop the panelists, made up of the presenters, plus event organizer Monroe Newborn and Chairman Tony Marsland, fielded questions from the audience. A sprited exchange took place as clarifications were requested and points made. Several good, thought provoking questions were raised, but as always the most heated remarks were about when and whether the performance of computers can be compared to that of the best humans. It is acknowledged that humans often make trivial mistakes (some passing unnoticed during play), while computers are prone to seemingly weak and ineffectual moves when at a loss for ideas. In the end there was no real concensus, although a sense of inevitability remained.

To close the session the moderator invited contribution to the "New Directions in Game-Tree Search" workshop, being held in conjunction with the 6th World Computer Chess Championship, Edmonton, Canada, 28 May to 1 June 1989.

References

1. T. Anantharaman, M. Campbell and F. Hsu, ‘‘Singular Extensions: Adding Selectivity to Brute-Force Searching,’’ *ICCA Journal* **11**(4), 135-143 (1988).

2. T. A. Marsland and F. Popowich, ‘‘Parallel Game-Tree Search,’’ *IEEE Trans. on Pattern Anal. and Mach. Intel.* **7**(4), 442-452 (1985).

3. T. A. Marsland, ‘‘Workshop Report: Theory and Practice in Computer Chess,’’ *ICCA Journal* **10**(4), 205-210 (1987).

4. M. Newborn, ‘‘Unsynchronized Iteratively Deepening Parallel Alpha-Beta Search,’’ *IEEE Trans. on Pattern Anal. and Mach. Intel.* **10**(5), 687-694 (1988).

5. J. Schaeffer, "Improved Parallel Alpha-Beta Search," *Procs. ACM/IEEE Fall Joint Comp. Conf.*, Dallas, Nov. 1986, 519-527.

6. T. Warnock and B. Wendroff, ‘‘Search Tables in Computer Chess,’’ *ICCA Journal* **11**(1), 10-13 (1988).